MOPV045

Data-Centric Web Infrastructure for CERN Radiation and Environmental Protection Monitoring

Adrien Ledeul, Catalina Cristina Chiriac, Gonzalo de la Cruz, Gustavo Segura, Jan Sznajd, CERN, Geneva, Switzerland

ICALEPCS 2021

Data-Centric Architecture for SCADA data exploitation

Objective

Provide comprehensive yet accessible means to extract and exploit SCADA generated data.

Data-Centric Mindset

The data-centric mindset considers the **data** to be the **permanent assets**, and the applications the temporary ones. The key principles, as expressed in the **Data-Centric Manifesto**, can be summarized as follows:

- Data is the key asset.
- Data is self-describing.
- Data is stored in non-proprietary formats.
- Data access control and security is the responsibility of the data layer itself.

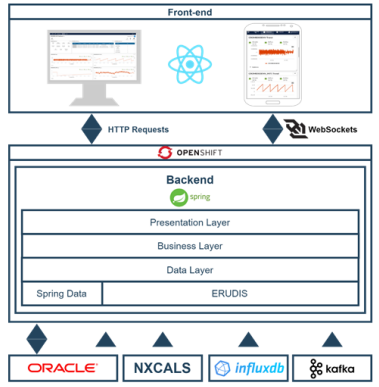
This approach suits well CERN's Radiation and Environment Monitoring Unified Supervision (**REMUS**), as data must be kept indefinitely.

Data-Centric principles applied to REMUS: The center of the diagram represents all the data associated with radiation and environmental protection monitoring at CERN. The external blocks represents applications consuming and producing data through APIs.

REMUS Web Architecture

Architecture Summary

- REMUS Web is deployed on the *CERN Platform-as-a-Service (PaaS) infrastructure*, based on *RedHat OpenShift*.
- The **front-end** runs on the user's web browser and communicates with the back-end via:
  - WebSockets for real-time communication.
  - HTTP requests for standard communication.
- The **back-end** is based on Spring Boot and follows a classic three-layer architecture, with *Presentation*, *Business* and *Data* layers.
  - The Data Layer is based on:
    - *Spring Data* for accessing REMUS Web entities and the SCADA metadata.
    - *ERUDIS* for accessing both historical and near-real-time data from the SCADA's instrumentation.



REMUS Web Architecture

REMUS Data Pipeline

The Lambda Architecture

REMUS Data pipeline follows the Lambda Architecture:

Batch processing

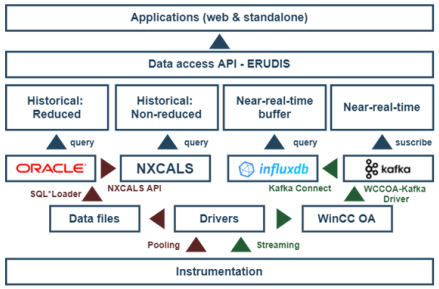
- Pooling from instrument's internal memories.
- Batch data file injection with *SQL*Loader*.
- Batch transfer to *Next CERN Accelerator Logging Service* (NXCALS).

Stream processing

- Stream through *Kafka*, via *WinCC OA*.
- *InfluxDB* is used for temporary data retention.

API

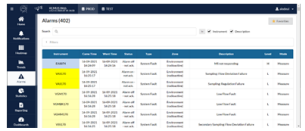
- *Radiation Unified Data Integration Service* (ERUDIS), based on *Akka* and *Alpakka*, unifies the access to the various data sources.



REMUS Data Pipeline: Applications consume data aggregated by the data access API. Data is fetched from various data sources with different latencies and time resolutions. Oracle and NXCALS are used for historical data; Kafka and InfluxDB for near-real-time data.

REMUS Web Functionalities

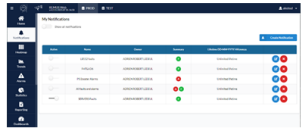
Near-real-time Trends: REMUS Web allows to access and plot both historical and near-real-time SCADA data. Users can fetch data from multiple data sources and define various parameters on the data to select.



Near-real-time Alarm Screen: REMUS alarm screen is accessible directly from the web and displays the generated alarms in near-real-time.

Dashboards: REMUS Web includes a powerful tool for creating and visualizing dashboards. Users can compose their own dashboards by combining different types of widgets. In addition, REMUS Web automatically generates a dashboard for every instrument connected to REMUS.

Metadata Statistics: REMUS Web allows access to the complete inventory of REMUS entities. Users can filter the data using multiple criteria and perform data aggregations for statistics extraction.



Notification Configuration: Users can configure personalized e-mail and SMS notifications based on events produced by the SCADA.

Domain-specific reports: REMUS users require domain-specific reports for radiation and environmental protection. REMUS Web includes tools to generate these reports.

Data-Centric Architecture for SCADA data exploitation

Objective

Provide comprehensive yet accessible means to extract and exploit SCADA generated data.

Data-Centric Mindset

The data-centric mindset considers the **data** to be the **permanent assets**, and the applications the temporary ones. The key principles, as expressed in the **Data-Centric Manifesto**, can be summarized as follows:

- Data is the key asset.
- Data is self-describing.
- Data is stored in non-proprietary formats.
- Data access control and security is the responsibility of the data layer itself.

This approach suits well CERN's Radiation and Environment Monitoring Unified Supervision (**REMUS**), as data must be kept indefinitely.

Data-Centric principles applied to REMUS: The center of the diagram represents all the data associated with radiation and environmental protection monitoring at CERN. The external blocks represents applications consuming and producing data through APIs.

REMUS Data Pipeline

The Lambda Architecture

REMUS Data pipeline follows the Lambda Architecture:

Batch processing

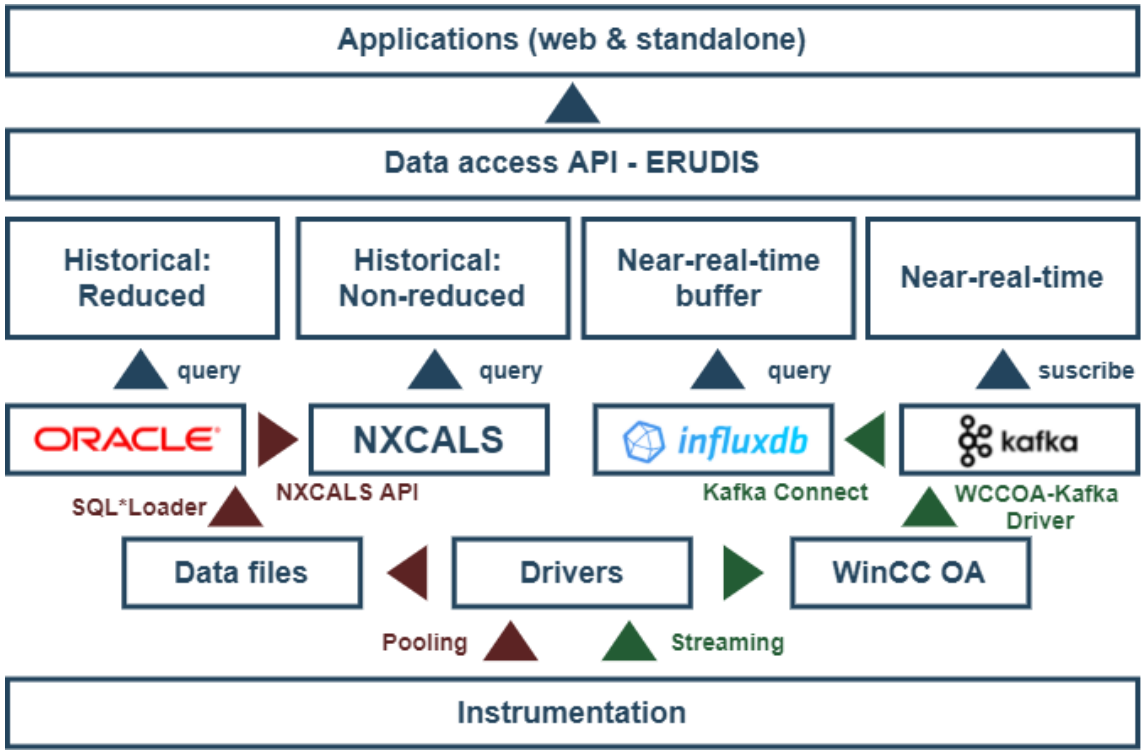
- Pooling from instrument's internal memories.
- Batch data file injection with *SQL*Loader.*
- Batch transfer to *Next CERN Accelerator Logging Service* (NXCALS).

Stream processing

- Stream through *Kafka*, via *WinCC OA.*
- *InfluxDB* is used for temporary data retention.

API

- *Radiation Unified Data Integration Service* (ERUDIS), based on *Akka* and *Alpakka*, unifies the access to the various data sources.

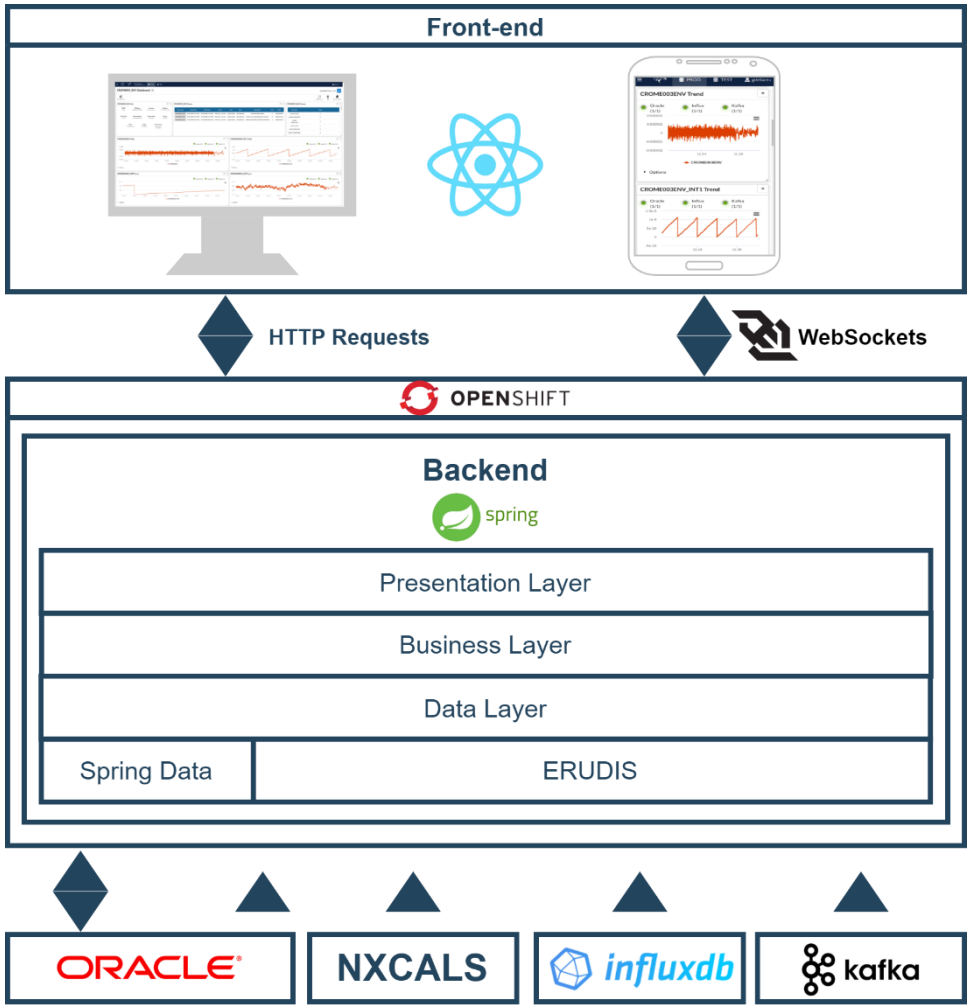


REMUS Data Pipeline: Applications consume data aggregated by the data access API. Data is fetched from various data sources with different latencies and time resolutions. Oracle and NXCALS are used for historical data; Kafka and InfluxDB for near-real-time data.

REMUS Web Architecture

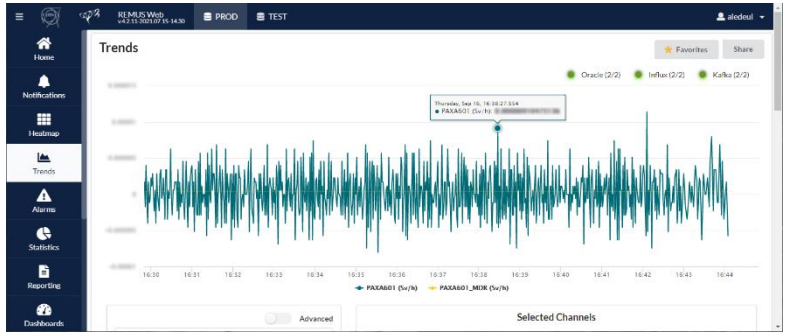
Architecture Summary

- REMUS Web is deployed on the *CERN Platform-as-a-Service (PaaS) infrastructure*, based on *RedHat OpenShift*.
- The **front-end** runs on the user's web browser and communicates with the back-end via:
  - WebSockets for real-time communication.
  - HTTP requests for standard communication.
- The **back-end** is based on Spring Boot and follows a classic three-layer architecture, with *Presentation*, *Business* and *Data* layers.
  - The Data Layer is based on:
    - *Spring Data* for accessing REMUS Web entities and the SCADA metadata.
    - *ERUDIS* for accessing both historical and near-real-time data from the SCADA's instrumentation.

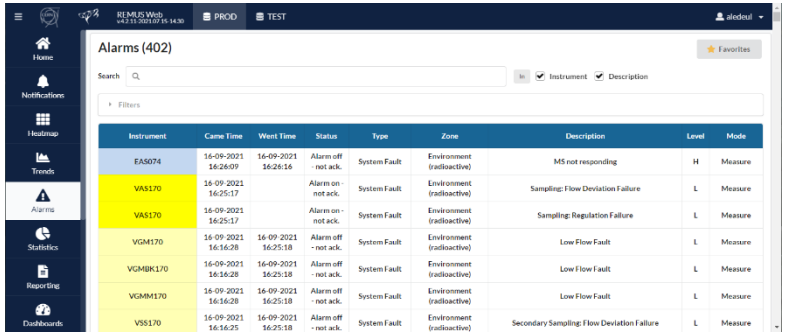


REMUS Web Architecture

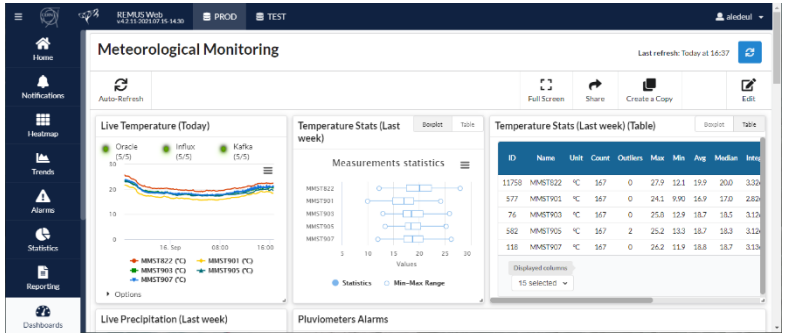
REMUS Web Functionalities



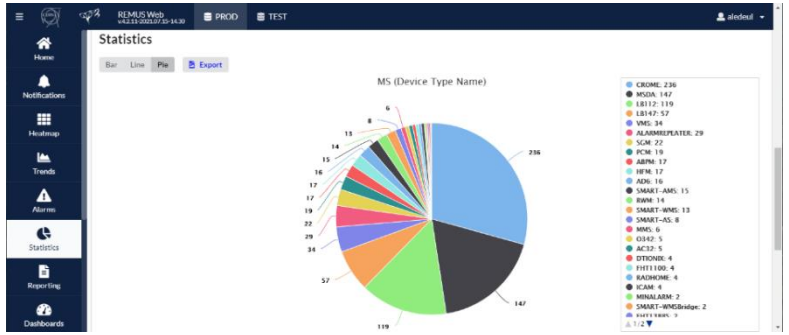
Near-real-time Trends: REMUS Web allows to access and plot both historical and near-real-time SCADA data. Users can fetch data from multiple data sources and define various parameters on the data to select.



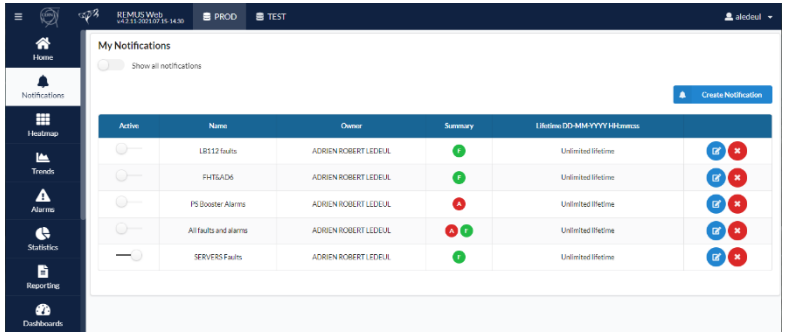
Near-real-time Alarm Screen: REMUS alarm screen is accessible directly from the web and displays the generated alarms in near-real-time.



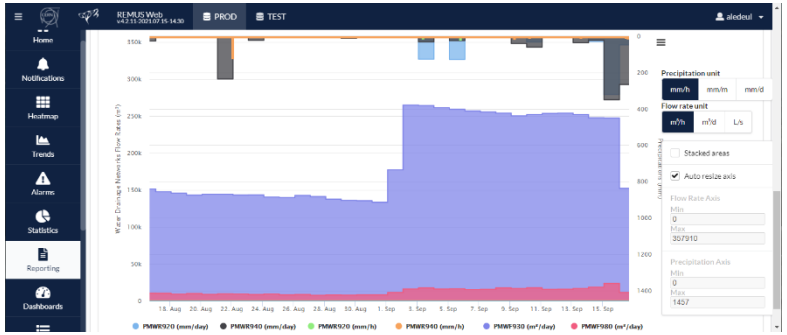
Dashboards: REMUS Web includes a powerful tool for creating and visualizing dashboards. Users can compose their own dashboards by combining different types of widgets. In addition, REMUS Web automatically generates a dashboard for every instrument connected to REMUS.



Metadata Statistics: REMUS Web allows access to the complete inventory of REMUS entities. Users can filter the data using multiple criteria and perform data aggregations for statistics extraction.



Notification Configuration: Users can configure personalized e-mail and SMS notifications based on events produced by the SCADA.



Domain-specific reports: REMUS users require domain-specific reports for radiation and environmental protection. REMUS Web includes tools to generate these reports.